CLAIMS

What is claimed is:

1. A method for efficient packet desegmentation on a network adapter, comprising:

buffering a plurality of data packet segments received at a network adapter from a single connection, wherein said single connection is identified by a plurality of addresses and ports extracted from each header of each of said plurality of data packet segments; and

responsive to detecting a buffering release condition, releasing said plurality of data packet segments from said network adapter as a desegmented group to a network stack, such that data packets segments received from said single connection are efficiently passed to said network stack together.

2. The method according to claim 1 for efficient packet desegmentation further comprising:

responsive to receiving a new data packet segment at said network adapter, extracting a plurality of addresses and ports for a connection across which said new data packet segment was sent;

responsive to said plurality of addresses and ports for said connection matching a buffered plurality of addresses and ports for said single connection, buffering said new data segment at said network adapter with said plurality of data packet segments previously buffered.

3. The method according to claim 1 for efficient packet desegmentation wherein said single connection is a TCP connection identified by a four-tuple of source and destination addresses and ports extracted from each TCP header of each of said plurality of data packet segments.

4. The method according to claim 1 for efficient packet desegmentation further comprising:

detecting said buffering release condition when a new data packet segment received at said network adapter is from a different connection than said single connection.

5. The method according to claim 1 for efficient packet desegmentation further comprising:

detecting said buffering release condition when a time a first receiving data packet segment from among said plurality of data packet segments is buffered at said network adapter exceeds a time threshold.

6. The method according to claim 1 for efficient packet desegmentation further comprising:

detecting said buffering release condition when a queue size limit in said network adapter for buffering data packet segments is reached.

7. The method according to claim 1 for efficient packet desegmentation further comprising:

detecting said buffering release condition when an abnormal condition occurs, wherein said abnormal condition is at least one from among a checksum mismatch, a connection reset, an urgent pointer, and a missing packet being detected.

8. A system for efficient packet desegmentation on a network adapter, comprising:

a network adapter with an interface for facilitating transfer of data packets between a data processing system and a network;

said network adapter further comprising:

a buffer for buffering a plurality of data packet segments received at said network adapter from a single connection across said network, wherein said single connection is identified by a plurality of addresses and ports extracted from each header of each of said plurality of data packet segments; and

a desegmenting means for releasing said plurality of data packet segments from said buffer together in a desegmented group to a network stack in said data processing system, responsive to detecting a buffering release condition.

9. The system according to claim 8 for efficient packet desegmentation, said desegmenting means further comprising:

means, responsive to receiving a new data packet segment at said network adapter, for extracting a plurality of addresses and ports for a connection across which said new data packet segment was sent;

means, responsive to said plurality of address and ports for said connection matching a buffered plurality of addresses and ports for said single connection, for buffering said new data segment in said buffer.

10. The system according to claim 8 for efficient packet desegmentation wherein said single connection is a TCP connection identified by a four-tuple of source and destination addresses and ports extracted from each TCP header of each of said plurality of data packet segments.

11. The system according to claim 8 for efficient packet desegmentation, said desegmenting means further comprising:

means for detecting said buffering release condition when a new data packet segment received at said network adapter is from a different connection than said single connection.

12. The system according to claim 8 for efficient packet desegmentation, said desegmenting means further comprising:

means for detecting said buffering release condition when a time a first receiving data packet segment from among said plurality of data packet segments remains within said buffer exceeds a time threshold.

13. The system according to claim 8 for efficient packet desegmentation, said desegmenting means further comprising:

means for detecting said buffering release condition when a queue size limit in said buffer is reached.

14. The system according to claim 8 for efficient packet desegmentation, said desegmenting means further comprising:

means for detecting said buffering release condition when an abnormal condition occurs, wherein said abnormal condition is at least one from among a checksum mismatch, a connection reset, an urgent pointer, and a missing packet being detected.

15. A computer program product for efficient packet desegmentation on a network adapter, comprising:

a recording medium;

means, recorded on said recording medium, for buffering a plurality of data packet segments received at a network adapter from a single connection, wherein said single connection is identified by a plurality of addresses and ports extracted from each header of each of said plurality of data packet segments; and

means, recorded on said recording medium, for releasing said plurality of data packet segments from said network adapter in a single desegmented group to a network stack, responsive to detecting a buffering release condition.

16. The computer program product according to claim 15 for efficient packet desegmentation further comprising:

means, recorded on said recording medium, for extracting a plurality of addresses and ports for a connection across which said new data packet segment was sent, responsive to receiving a new data packet segment at said network adapter;

means, recorded on said recording medium, for buffering said new data segment at said network adapter with said plurality of data packet segments previously buffered, responsive to said plurality of addresses and ports for said connection matching a buffered plurality of addresses and ports for said single connection.

17. The computer program product according to claim 15 for efficient packet desegmentation further comprising:

means, recorded on said recording medium, for detecting said buffering release condition when a new data packet segment received at said network adapter is from a different connection than said single connection.

18. The computer program product according to claim 15 for efficient packet desegmentation further comprising:

means, recorded on said recording medium, for detecting said buffering release condition when a time a first receiving data packet segment from among said plurality of data packet segments is buffered at said network adapter exceeds a time threshold.

19. The computer program product according to claim 15 for efficient packet desegmentation further comprising:

means, recorded on said recording medium, for detecting said buffering release condition when a queue size limit in said network adapter for buffering data packet segments is reached.

20. The computer program product according to claim 15 for efficient packet desegmentation further comprising:

means, recorded on said recording medium, for detecting said buffering release condition when an abnormal condition occurs, wherein said abnormal condition is at least one from among a checksum mismatch, a connection reset, an urgent pointer, and a missing packet being detected.